

*Una buena ingeniería de software
requiere la diferenciación entre la
especificación y la implementación.*

INGENIERIA DE SOFTWARE

Primer Avance

García Juárez Leonel Alexis

Medina Avendaño Griselda

Ramírez Ramírez Ramiro

Índice general

0.1. Introducción	4
1. Propósito	5
2. Ámbito del Proyecto	6
3. Personal Involucrado	7
4. Factibilidad	8
5. Mapa de Navegación	9
6. Requisitos Funcionales	10
7. Interfaces	20
8. Estándar de Codificación	29
8.1. Inicialización	29
8.2. Variables locales	29
8.3. Variables globales	29
8.4. Constantes	29
8.5. Comentarios de inicio o encabezado	30
8.6. Comentarios en línea	30
8.7. Sangrías y tabulaciones	30
8.8. Interfaces	30

Índice de figuras

3.1. Roles de usuario	7
5.1. Mapa de navegación	9
6.1. Requisito Funcional 1	10
6.2. Requisito Funcional 2	11
6.3. Requisito Funcional 3	12
6.4. Requisito Funcional 4	13
6.5. Requisito Funcional 5	14
6.6. Requisito Funcional 6	15
6.7. Requisito Funcional 7	16
6.8. Requisito Funcional 8	17
6.9. Requisito Funcional 9	18
6.10. Requisito Funcional 10	19
7.1. Login	20
7.2. Pantalla Principal	21
7.3. Nuevo Producto	22
7.4. Modificar Producto	23
7.5. Eliminar Producto	24
7.6. Inventario	25
7.7. Ventas	26
7.8. Corte	27
7.9. Reporte	28

.

0.1. Introducción

Este proyecto ha sido elaborado dada la necesidad de automatizar un proceso de ventas de una ferretería, la cual requiere generar informes exactos sobre las ventas diarias, además de que requiere de llevar un control de inventario de productos para poder supervisar el stock disponible de los mismos, además se pretende que a través de este sistema los empleados mejoren sus tiempos de atención al cliente en ventas ya que evitarían hacer cuentas de forma manual.

El sistema de software que implementaremos facilitará el manejo de ventas, inventario, ingreso de productos, etc ya que esto se realizará de manera sistematizada ya que actualmente se realiza de manera manual, por lo anterior de este sistema ayudaría a visualizar el rendimiento de la empresa tomando en cuenta las ventas generadas por el sistema.

Este proyecto será realizado con el lenguaje de programación Java, además de que teniendo en cuenta que es un sistema que va a generar una gran cantidad de datos debido a las diferentes funcionalidades del mismo, se pretende hacer uso de una base de datos que los almacene. Como se mencionó antes este sistema contará con distintas funcionalidades las cuales ayudarían a agilizar tareas como la generación de reportes de ventas por días, generación de corte diario, manejo de ventas por cliente, cobro por ventas, y el manejo del inventario.

La realización del estudio de métodos en la ferretería es importante debido a que partir de este, se puede realizar una mejora a los procesos estudiados, haciendo que sean más óptimos y que a su vez permitan aumentar los niveles de productividad de la compañía.

Capítulo 1

Propósito

El objetivo de este proyecto constituye una herramienta que permite un manejo oportuno y eficiente de la información, siendo posible el rendimiento de tiempo respecto a sus productos. Al igual que permita lograr la eficiencia en el control de los productos con los que se va a comercializar, como también llevar un control del inventario, registro de ventas en productos de una manera más rápida, organizada y sencilla beneficiando tanto al personal como a los clientes de la empresa. Emitirá reportes sobre ventas de manera eficaz, de forma que se logre un buen manejo de las transacciones que incurren en la empresa.

Capítulo 2

Ámbito del Proyecto

La ferretería a la cual está destinado el proyecto, desea un sistema de administración para ingreso y salida de mercancía sobre control del inventario y ventas, que ayude a mejorar la administración y control de la empresa, ya que al contar con un sistema manual se tienen dificultades para obtener de forma ágil y rápida la información relevante sobre las existencias. El proyecto consiste en diseñar y construir un sistema de interfaz amigable que permita llevar un control sobre los productos disponibles, además de mantener un registro de las salidas de los productos, de tal forma que esta información sirva al propietario del negocio para tomar decisiones como cuando reabastecerse de productos, también se pretende que el sistema gestione las ventas realizando su registro y generando un reporte de las mismas cuando este se requiera. Para el desarrollo de este proyecto se hará uso de distintas técnicas de programación aprendidas durante el curso como programación orientada a objetos, este paradigma nos servirá para la creación de la interfaz gráfica de usuario así como la funcionalidad deseada, combinando el lenguaje de programación Java con el lenguaje de consulta de mysql para poder manipular de mejor manera los datos.

Capítulo 3

Personal Involucrado

Nombre	Rol de usuario
Everardo de Jesus Pacheco Antonio	Administrador
Rolando Pedro Gabriel	Administrador
Lirio Ruiz Guerra	Empleado
Ramiro Ramírez Ramírez	Administrador
Leonel Alexis García Juárez	Administrador
Griselda Medina Avendaño	Administrador

Figura 3.1: Roles de usuario

Capítulo 4

Factibilidad

Es posible realizar este proyecto dado que para la factibilidad técnica, el equipo de desarrollo cuenta con conocimientos en programación en java así como el manejo de base de datos, además de que se cuenta con herramientas como el entorno de desarrollo Netbeans de cual se hará uso, además del sistema gestor de base de datos Postgres. En cuanto a la factibilidad operativa, se cuenta con usuarios finales dispuestos a emplear las funcionalidades generadas por el sistema que se desarrollará.

Capítulo 5

Mapa de Navegación

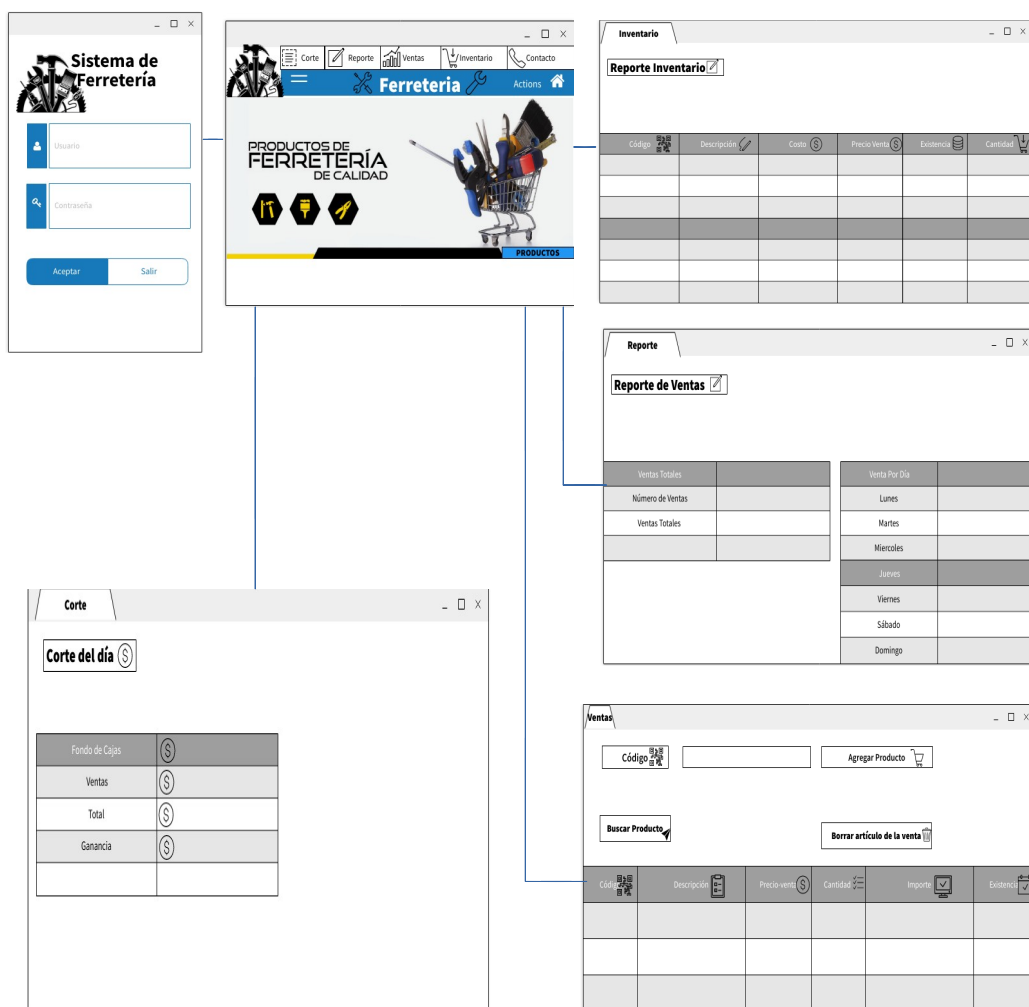


Figura 5.1: Mapa de navegación

Capítulo 6

Requisitos Funcionales

IDENTIFICADOR RF1		NOMBRE Buscar Producto	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Producto		SALIDA Producto con todos sus datos (descripción, precio, existencia)	
DESCRIPCIÓN PRECONDICIÓN: Se desea localizar un producto por nombre o código. DESCRIPCIÓN: Se verifica el nombre del producto y el código. POSTCONDICIÓN: Se localiza el producto y se muestran sus datos.			
MANEJO DE SITUACIONES ANORMALES 1. Producto inexistente			
CRITERIOS DE ACEPTACIÓN 1. EL código ingresado es correcto, se encuentra el producto.			

Figura 6.1: Requisito Funcional 1

IDENTIFICADOR RF2		NOMBRE Agregar Producto a la venta	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código		SALIDA Se agrega producto a la venta	
DESCRIPCIÓN PRECONDICIÓN: Se requiere localizar producto para agregar a la venta. DESCRIPCIÓN: Se busca producto deseado POSTCONDICIÓN: Se agrega producto localizado a la venta.			
MANEJO DE SITUACIONES ANORMALES 1.Producto inexistente, se envía mensaje de error al usuario.			
CRITERIOS DE ACEPTACIÓN 1. EL código ingresado es correcto, se localiza el producto.			

Figura 6.2: Requisito Funcional 2

IDENTIFICADOR RF3		NOMBRE Borrar articulo de la venta	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Ventas		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Selección del producto		SALIDA Producto eliminado de la venta	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con un producto de la venta seleccionado. DESCRIPCIÓN: Se consulta al usuario si quiere eliminar el artículo. POSTCONDICIÓN: Se elimina el producto de la venta.			
MANEJO DE SITUACIONES ANORMALES			
CRITERIOS DE ACEPTACIÓN 1. Se selecciona producto de la venta y se elimina de la misma.			

Figura 6.3: Requisito Funcional 3

IDENTIFICADOR RF4		NOMBRE Nuevo Producto	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Descripción Cantidad Precio costo Precio venta		SALIDA Alta del producto en el sistema	
DESCRIPCIÓN PRECONDICIÓN: No se cuenta con registro del producto en sistema. DESCRIPCIÓN: Se ingresan los datos del producto. POSTCONDICIÓN: Producto es dado de alta en el sistema.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error y no se reconoce el nuevo producto).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se agrega un nuevo producto.			

Figura 6.4: Requisito Funcional 4

IDENTIFICADOR RF5		NOMBRE Modificar Producto	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código Descripción Precio Costo Precio venta Cantidad		SALIDA Datos de producto son modificados	
DESCRIPCIÓN PRECONDICIÓN: Se desea modificar algún dato del producto. DESCRIPCIÓN: Se ingresan los datos y el código del producto. POSTCONDICIÓN: Se modifica el producto.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se modifica el producto.			

Figura 6.5: Requisito Funcional 5

IDENTIFICADOR RF6		NOMBRE Eliminar Producto	
TIPO (NECESARIO/DESEABLE) Deseable	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección productos		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre del producto Código		SALIDA Producto eliminado del sistema	
DESCRIPCIÓN PRECONDICIÓN: Se desea dar de baja un producto, eliminar todos sus datos. DESCRIPCIÓN: Se ingresa código o nombre del producto para su localización. POSTCONDICIÓN: Se elimina producto del inventario.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se elimina el producto.			

Figura 6.6: Requisito Funcional 6

IDENTIFICADOR RF7		NOMBRE Reporte de inventario	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Inventario		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Productos almacenados en la base de datos		SALIDA Reporte de inventario.	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con productos dados de alta. DESCRIPCIÓN: Se ingresa a la sección inventario. POSTCONDICIÓN: Se genera un reporte de inventario donde se pueden ver todos los productos disponibles con sus datos correspondientes.			
MANEJO DE SITUACIONES ANORMALES 1. No se han dado de alta productos en sistema (Se mostrará en pantalla un mensaje de inexistentes).			
CRITERIOS DE ACEPTACIÓN 1. Existen productos dados de alta, se genera el reporte de inventario.			

Figura 6.7: Requisito Funcional 7

IDENTIFICADOR RF8		NOMBRE Reporte de ventas	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección reportes		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Ventas totales realizadas en una semana		SALIDA Reporte de: Ventas totales Número de ventas Venta promedio Ventas por día	
DESCRIPCIÓN PRECONDICIÓN: Se han realizado ventas DESCRIPCIÓN: Se consulta el importe total generado por cada venta. POSTCONDICIÓN: Se genera el reporte de ventas.			
MANEJO DE SITUACIONES ANORMALES 1. No se han realizado ninguna venta en la semana(Se mostrará en pantalla un mensaje).			
CRITERIOS DE ACEPTACIÓN 1. Se han realizado ventas, se genera el reporte de ventas.			

Figura 6.8: Requisito Funcional 8

IDENTIFICADOR RF9		NOMBRE Hacer corte del día	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección corte		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Importe total de las ventas realizadas en el día Fondo de caja inicial		SALIDA Visualización de: Fondo de caja Ventas Total Ganancia	
DESCRIPCIÓN PRECONDICIÓN: Se cuenta con ventas realizadas en el día. DESCRIPCIÓN: Se realizan cálculos para mostrar total de ventas y ganancias. POSTCONDICIÓN: Se genera el corte del día.			
MANEJO DE SITUACIONES ANORMALES 1. No se ha realizado ni una venta en el día (Se mostrará en pantalla un mensaje al usuario).			
CRITERIOS DE ACEPTACIÓN 1. Se han realizado ventas en el día, se genera corte del día.			

Figura 6.9: Requisito Funcional 9

IDENTIFICADOR RF10		NOMBRE Agregar Usuario	
TIPO (NECESARIO/DESEABLE) Necesario	REQUERIMIENTO QUE LO UTILIZA O ESPECIALIZA Sección Usuarios		CRÍTICO? Si
PRIORIDAD DE DESARROLLO Alta			
ENTRADA Nombre Contraseña Rol		SALIDA Registro de nuevo usuario.	
DESCRIPCIÓN PRECONDICIÓN: No se cuenta con registro de usuario en el sistema. DESCRIPCIÓN: Se ingresan los datos del usuario a registrar. POSTCONDICIÓN: Se registra el nuevo usuario.			
MANEJO DE SITUACIONES ANORMALES 1. Datos erróneos (Se mostrará en pantalla un mensaje de error y no se ingresa el usuario).			
CRITERIOS DE ACEPTACIÓN 1. Los datos ingresados son correctos, se registra un nuevo usuario.			

Figura 6.10: Requisito Funcional 10

Capítulo 7

Interfaces



The image shows a login window titled "Sistema de Ferretería". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. The main content area features a logo of various tools (hammer, wrench, saw, etc.) on the left and the title "Sistema de Ferretería" on the right. Below the logo, there are two input fields: "Usuario" (with a person icon) and "Contraseña" (with a key icon). At the bottom, there are two buttons: "Aceptar" (Accept) and "Salir" (Exit).

Figura 7.1: Login



Figura 7.2: Pantalla Principal

Nuevo Producto

Nuevo

Modificar

Eliminar

Nuevo Producto

Código

Descripción

Precio Costo

Precio Venta

Cantidad

Guardar Producto

\$

\$

Figura 7.3: Nuevo Producto

Modificar Producto

Nuevo

Modificar

Eliminar

Nuevo Producto

Código

Descripción

Precio Costo

Precio Venta

Cantidad

Guardar Producto

Aceptar

Figura 7.4: Modificar Producto

Eliminar Producto

Nuevo

Modificar

Eliminar

Nuevo Producto

Código

Descripción

Precio Costo

Precio Venta

Cantidad

Eliminar Producto

Figura 7.5: Eliminar Producto

Inventario

Reporte Inventario

Código	Descripción	Costo	Precio Venta	Existencia	Cantidad

Figura 7.6: Inventario

Ventas

Código

Agregar Producto

Buscar Producto

Borrar artículo de la venta

Código	Descripción	Precio-venta	Cantidad	Importe	Existencia

Figura 7.7: Ventas

Corte	
Corte del día \$	
Fondo de Cajas	\$
Ventas	\$
Total	\$
Ganancia	\$

Figura 7.8: Corte

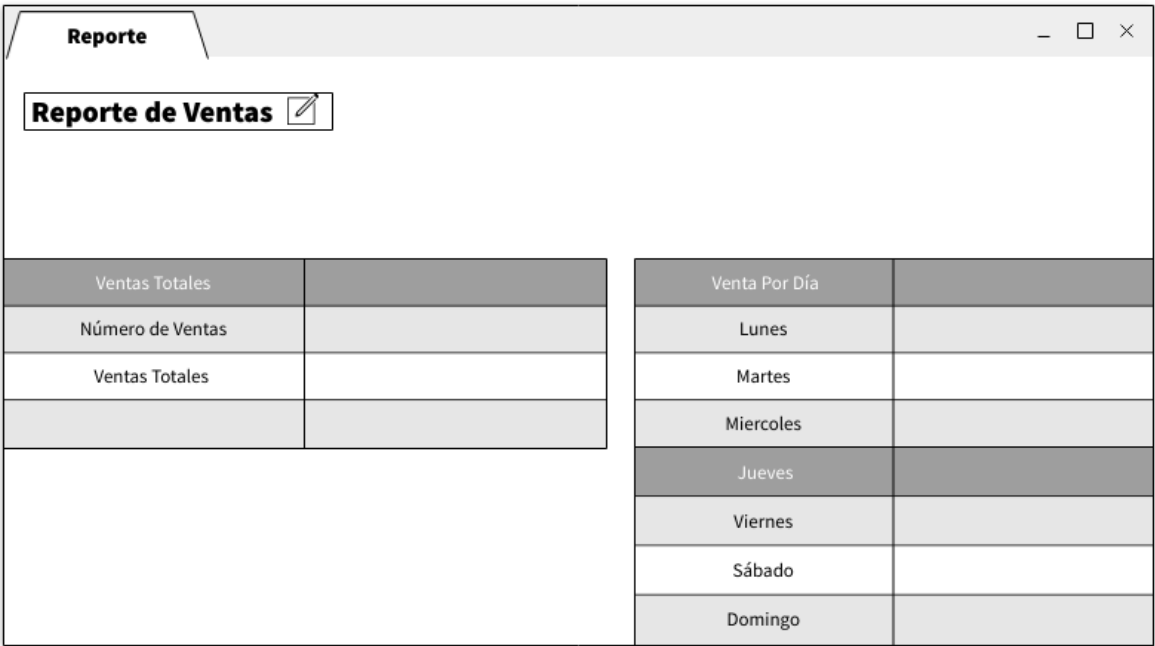


Figura 7.9: Reporte

Capítulo 8

Estándar de Codificación

[T1]fontenc[utf8]inputenc

8.1. Inicialización

Toda variable local deberá ser inicializada en el momento de declaración, deberá de estar situada al principio de cada bloque principal.

8.2. Variables locales

Deben estar situadas en el principio de cada bloque principal:

```
public class Ejemplo
    public static void main(String[] args)
        //Declaración de variables
        int contador=0;
        double r=0.0;
```

8.3. Variables globales

Estas deben estar ubicadas al inicio de la clase principal.

Notación

Se usará la notación CamelCase para la declaración de métodos o variables. //ejemplo de un método definido con CamelCase en CalcularSuma public int CalcularSuma(int b, int a) int resultado; resultado=b+a; return resultado;

8.4. Constantes

La declaración de contantes deberá ser declarado con letras mayúsculas.

```
public class Ejemplo
    public static void main(String[] args)
        //Declaración de constantes
        static final int DIASSEMANA = 7;
```

```
static final int DIASLABORABLES = 5;
```

8.5. Comentarios de inicio o encabezado

Se usará un encabezado al inicio de cada bloque o clase creada, también deberá ser asignada a las interfaces. El encabezado usado hará uso de comentarios en bloque como se muestra a continuación:

```
/** @Autor:                               */
/* Descripción del proyecto:             */
/* Fecha de creación:                    */
/* Fecha de modificación:                */
package ejemplo;
public class Ejemplo
public static void main(String[] args)
//Código
```

8.6. Comentarios en línea

Estos comentarios se deberán hacer en una línea, con cierto espacio si se quiere documentar la función de alguna operación o método.

```
public int CalcularSuma(int b, int a)
int resultado; //Se declara una variable de tipo entero
resultado=b+a; //Calcula la suma de b+a y la guarda en la variable resultado
return resultado; //Retorna la variable resultado
```

8.7. Sangrías y tabulaciones

Las tabulaciones serán igual a cuatro espacios, mientras que las sangrías deberán de mostrar claramente los procesos mas internos del código.

```
public class Ejemplo
    public static void main(String[] args)
        int num=0;
        if(num==0)
            System.out.println("Número igual a cero");

        else if(num>0)
            System.out.println("Número mayor que cero");

        else
            System.out.println("Número menor que cero");
```

8.8. Interfaces

Para los nombres que serán usados en los componentes de las interfaces se agregará un identificador, esto con el fin de reconocer mas fácil a los componentes. Los componentes que se usaran serán los

siguientes:

```
jTextField txtEjemplo  
jLabel lblEjemplo  
jButton btnEjemplo  
jList lsEjemplo  
jRadioButton rbtnEjemplo  
jCheckBox cbEjemplo  
jComboBox cbxEjemplo  
jPanel jpEjemplo  
jTabbedPane jtEjemplo  
jPasswordField jpsEjemplo  
jTable tbEjemplo
```